User's guide: Manual for V-Xtractor 2.1

This is a guide to install and use the software utility V-Xtractor. The software is reasonably platform-independent. The instructions below should work fine with little or no modifications for nearly all UNIX-type systems including MacOS X. Windows users are advised to install Cygwin (http://www.cygwin.com) in order to compile and run the utilities. Please read chapter 5 carefully for important information about this software.

1.   The main principle

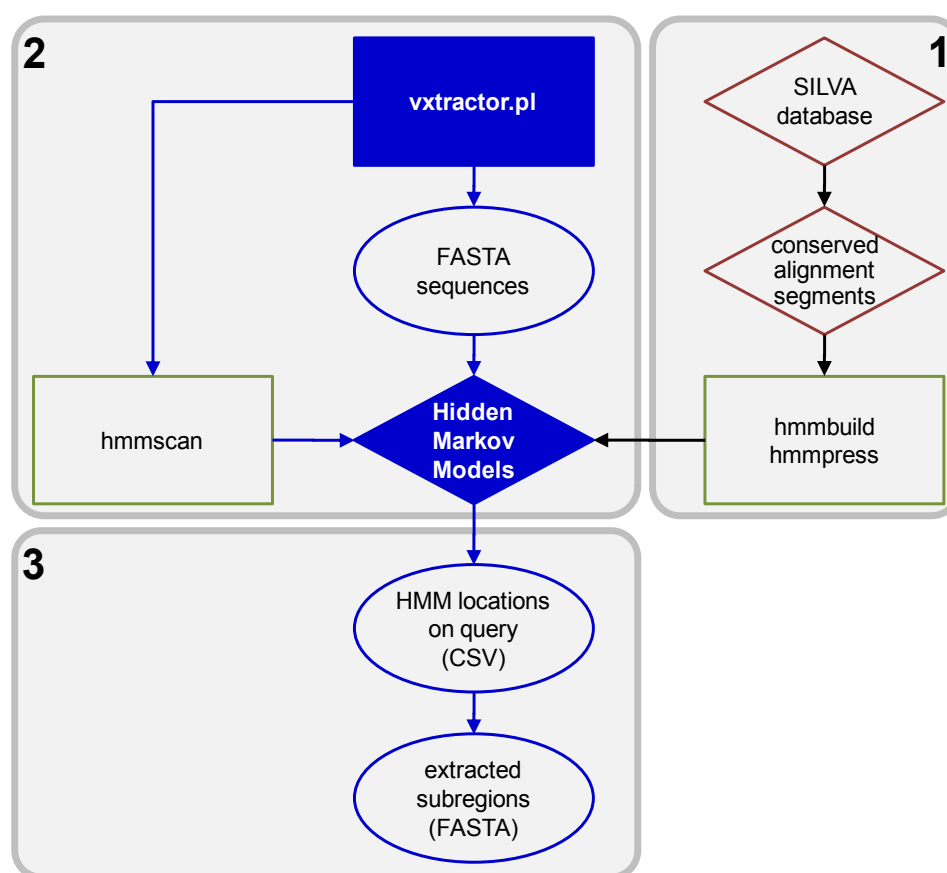Figure 1 shows the workflow of V-Xtractor and how the utility interacts with HMMER.

Figure 1. The V-Xtractor workflow. The different steps are explained in the main text. Functions performed by V-Xtractor are indicated in blue, function by HMMER in green, and data retrieved from the SILVA database in red. Utilities of the V-Xtractor package are indicated by filled blue boxes and include the actual script vxtractor.pl and the Hidden Markov Models.

Part 1: Alignment segments representing the conserved boundaries of all hypervariable regions along the small- or large-subunit ribosomal RNA gene ("V" and "D" regions, respectively) were extracted from the SILVA reference alignment. The HMMER programs hmmbuild and hmmpress were used to generate the Hidden Markov Models (HMMs) from these alignment segments. The HMMs are included in the V-Xtractor package, and the SILVA database or any other parts of this first step are **not** required in order to run V-Xtractor.

Part 2: The script vxtractor.pl calls the HMMER program hmmscan and runs each HMM against each query sequence in the FASTA file. Predefined detection thresholds that are hardcoded in the HMM files are used to define true targets.

Part 3: Location of each HMM on each sequence or failure of its detection is reported in a comma-separated value (csv) file. Based on the HMM location, the target region defined by the user is extracted from the query file and stored in a new output FASTA file ready for downstream analysis.

2. Detailed installation instructions

The README.txt file bundled with the script provides a quick installation guide. The instruction below are more detailed and intended for users with less experience in compiling software. In order to install certain packages, you might need to have superuser privileges (sudo password on MacOS; root password on Linux).

2.1 For installation on Mac, you will have to install the Apple Xcode package available on your MacOS X System DVD in order to be able to compile programs. See http://developer.apple.com/mac/library/documentation/Xcode/Conceptual/XcodeCoexistence/Contents/Resources/en.lproj/Basics/Basics.html for detailed installation instructions. Please talk to your system administrator if you feel unsure about these steps. Note that the packages are mandatory and that you should not proceed unless these criteria are fulfilled.

2.2 Perl needs to be installed on the computer. Most Unix-based systems including Linux and MacOS X have Perl pre-installed. You can check this by opening a command line terminal and type "perl -v". In case Perl is not installed you have to download (http://www.perl.org) and compile the program. Windows users are advised to compile Perl with Cygwin or, alternatively, install ActivePerl (http://www.activestate.com/activeperl/).

2.3 Download and install HMMER version 3

The current version of V-Xtractor relies on HMMER version 3. (An older version for HMMER version 2 can be requested from the authors.) Windows users can compile HMMER with Cygwin. The following 8 steps will install HMMER on your computer.

2.3.1 Go to http://hmmer.janelia.org/#download and download the HMMER source code or the precompiled binary for your operating system (if available; (in this example, HMMER is compiled form source code)) to any preferred directory, e.g. your home directory "/home/username/" (on Linux) or "/Users/username/" (on MacOS). For example if your username is "martin", this would be "/home/martin/" on Linux or "/Users/martin/" on MacOS respectively. We will use the directory "/home/martin/" for all further instructions.

2.3.2 Open a command line terminal, move into the directory with the change directory command "cd /home/martin/". Note that there is always a space between the command (in this case "cd") and the path.

2.3.3 Unpack the tarball with "tar zxf hmmer-3.0.tar.gz". Note that the package might have a different name than hmmer-3.0.tar.gz depending on what version of the package was downloaded. You can type the list command "ls" in your terminal in order to display the name of the file or look it up in your file browser.

2.3.4 Once unpacked, enter the new directory with "cd hmmer-3.0/" command. Again, the name of the directory might be different depending on what package you have downloaded.

2.3.5 Type "./configure" followed by Enter.

2.3.6 Type "make" followed by Enter. HMMER is now compiled but not installed in your home path, which will be done in the next step.

2.3.7 For this step you have to be logged in as superuser.

On Linux, type "su" followed by Enter. You will be prompted for the root (superuser) password. Type the password followed by Enter to log in as superuser. Then type "make install" followed by Enter in order to install HMMER at a location where your path reaches it. Log off from superuser by typing "exit" followed by Enter.

On MaxOS X, type "sudo make install" followed by Enter. You will be prompted for your regular system password. Enter your password followed by Enter; HMMER will now be installed where reachable by your path such that you do not need to worry where the binaries are located.

2.3.8 The HMMER package should have been compiled and installed on your computer; you can check this by typing "hmmscan -h" in the terminal and press Enter; you should now see the HMMER output. This step must work for V-Xtractor to function properly. The HMMER programs can be executed from any location on the computer and you do not have to put V-Xtractor in the same directory.

2.4 Download and unpack V-Xtractor

Go to http://www.cmde.science.ubc.ca/mohn/software.html in order to download the software package called vxtractor.zip and save it to your directory "/home/martin/". In the command line terminal, move into the directory with "cd /home/martin/" and extract the zip file with "unzip vxtractor.zip". A directory called "vxtractor" will be created. Move into the directory with "cd vxtractor/" and type "ls"; this will list following files and directories, vxtractor.pl (the actual script), the GPL license, the HMMs directory (containing the Hidden Markov Models), the User's Guide, the README.txt file as well as test input and output files. Now you are ready to use V-Xtractor - no further installation steps are required.

3. Usage and commands:

Copy your query FASTA file, from which you want to extract the sequence sub-regions, to the V-Xtractor directory; in our case this is "/home/martin/vxtractor/". Move into this directory using the command line terminal and "cd /home/users/vxtractor/" and type "perl vxtractor.pl". This will print all command options on the screen (Figure 2). All options and specifications are listed in Table 1 below. In order to perform a test run, you can use the testfileSSUbact.fasta that comes bundled with the script. This file contains 100 randomly selected entries of the SILVA bacterial SSU reference alignment. The two output files generated by the script are also included for your convenience. Start an extraction process by typing the command below followed by Enter; the extraction process will begin.

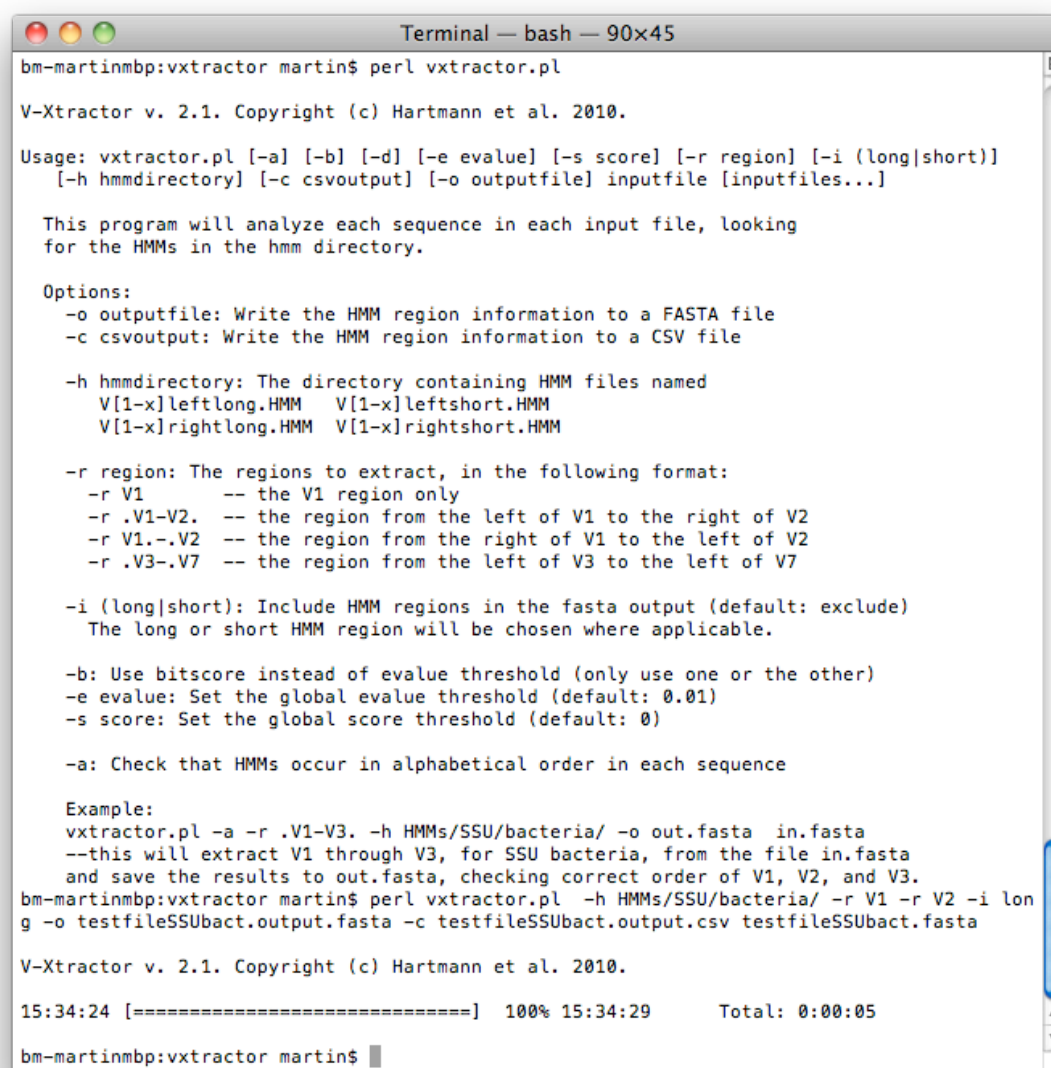**perl vxtractor.pl -h HMMs/SSU/bacteria/ -r V1 -r V2 -i long -o testfile.output.fasta -c testfile.output.csv testfile.fasta**

This will make V-Xtractor use the bacterial SSU HMMs (-h HMMs/SSU/bacteria/) to extract the V1 and V2 region (-r V1 -r V2). The option -i long specifies that the region matched to the HMMs should be included in the output sequence; -o is used to specify the name of the FASTA output file; and -c is used to specify the name of the comma separated output file. The name of the input file is testfileSSUbact.fasta.

Table 1. List of commands for V-Xtractor.

Commands	Description
input file	Name of input FASTA file.
-o	Name of output FASTA file. The FASTA header contains the following information: Sequence name derived from input file; name of extracted region; position of the extracted region on the query sequence; information about co-extraction of the HMM region (see option -i for details); error messages for HMMs with non-fatal problems (see -c for details).
-h HMM/SSU/bacteria HMM/SSU/archaea HMM/SSU/fungi HMM/LSU/bacteria HMM/LSU/archaea HMM/LSU/fungi	Name of directory for Hidden Markov Models, e.g. HMMs/SSU/bacteria. The user can specify new directories for other HMMs. Files are named V[1-x]leftlong.HMM V[1-x]leftshort.HMM V[1-x]rightlong.HMM V[1-x]rightshort.HMM Note: If more than nine regions are targeted, rename the HMMs from V1-V9 to V01-V09 in order to maintain the operability of option -a. The program is general enough that as long as the suffix and extension are present, the name of each region is unimportant.
-c	Name of output table containing information about detection and position of the HMMs for every query sequence. HMMs that are not detected are marked with "notfound". In addition, the absolute position of every region is listed. Irregular cases are marked with special codes. HMM= shortleft only the short HMM was detected at the left boundary of the region HMM= longright only the long HMM was detected at the right boundary of the region HMM= mismatchleft short HMM is located outside (left) of long HMM
-r	Specifies target region to be extracted. All hypervariable regions are extracted by default. The following syntax should be used (dots indicate start and stop positions). V1 the respective V-region only (V1left.HMM to V1right.HMM) .V1-V2. region left of V1 to right of V2 (V1left.HMM to V2right.HMM) V1.-.V2 region right of V1 to left of V2 (V1right.HMM to V2left.HMM) .V3-.V7 region left of V3 to left of V7 (V3left.HMM to V7left.HMM)
-i long short	Specifies whether the HMM region will be included in the extracted sequence (excluded by default, i.e. starts 1bp downstream of HMMleft and stops 1bp upstream of HMMright). The user can define if long or short HMM regions are included where applicable. If the chosen HMM size is not found, the other HMM is used and the substitution is marked in the FASTA header.
-b	*hmmscan* can use only one threshold parameter, i.e. e-value or score. By default, vxtractor.pl uses the e-value. The user can switch to using the score by selecting this function.
-e **-s**	Sets the *hmmscan* default threshold for e-value (-e) and score (-s), for HMM files that do not have these values set internally. The bundled HMM files have these values hardcoded (see line starting with "DESC"). The internally set values were tailored for optimal performance. Changing thresholds is useful when testing new HMMs not containing tailored thresholds.
-a	This option will check for the correct order of the regions and report sequence anomalies in the csv file (-c). The script automatically checks that HMMleft comes upstream of HMMright for a given region. For SSU sequences for example, option -a will check that V1, V2, V3…V9 appear in the correct order on the query sequence. Note: As long as the left|right, long|short suffixes and HMM extensions are present, any naming scheme can be used. The program sorts them alphabetically, in an effort to be as general as possible (see renaming convention under option –h for details).

4. Interpretation of the output

A progress bar on the screen indicates when the process is expected to be finished (see bottom of screenshot Figure 2). Detection failure or anomalies will be reported on the screen. Once the execution of the script has been completed, one or two files will be found in the current directory, i.e. the FASTA file and/or comma-separated value (.csv) file.


```
bm-martinmbp:vxtractor martin$ perl vxtractor.pl

V-Xtractor v. 2.1. Copyright (c) Hartmann et al. 2010.

Usage: vxtractor.pl [-a] [-b] [-d] [-e evalue] [-s score] [-r region] [-i (long|short)]
    [-h hmmdirectory] [-c csvoutput] [-o outputfile] inputfile [inputfiles...]

  This program will analyze each sequence in each input file, looking
  for the HMMs in the hmm directory.

  Options:
    -o outputfile: Write the HMM region information to a FASTA file
    -c csvoutput: Write the HMM region information to a CSV file

    -h hmmdirectory: The directory containing HMM files named
        V[1-x]leftlong.HMM   V[1-x]leftshort.HMM
        V[1-x]rightlong.HMM  V[1-x]rightshort.HMM

    -r region: The regions to extract, in the following format:
      -r V1      -- the V1 region only
      -r .V1-V2. -- the region from the left of V1 to the right of V2
      -r V1.-.V2 -- the region from the right of V1 to the left of V2
      -r .V3-.V7 -- the region from the left of V3 to the left of V7

    -i (long|short): Include HMM regions in the fasta output (default: exclude)
      The long or short HMM region will be chosen where applicable.

    -b: Use bitscore instead of evalue threshold (only use one or the other)
    -e evalue: Set the global evalue threshold (default: 0.01)
    -s score: Set the global score threshold (default: 0)

    -a: Check that HMMs occur in alphabetical order in each sequence

    Example:
    vxtractor.pl -a -r .V1-V3. -h HMMs/SSU/bacteria/ -o out.fasta  in.fasta
    --this will extract V1 through V3, for SSU bacteria, from the file in.fasta
    and save the results to out.fasta, checking correct order of V1, V2, and V3.
bm-martinmbp:vxtractor martin$ perl vxtractor.pl  -h HMMs/SSU/bacteria/ -r V1 -r V2 -i lon
g -o testfileSSUbact.output.fasta -c testfileSSUbact.output.csv testfileSSUbact.fasta

V-Xtractor v. 2.1. Copyright (c) Hartmann et al. 2010.

15:34:24 [==============================] 100% 15:34:29     Total: 0:00:05

bm-martinmbp:vxtractor martin$
```

Figure 2. Screenshots of V-Xtractor. Typing "perl vxtractor.pl" lists all command options on the screen. The command at the bottom shows an actual run of the script using testfileSSUbact.fasta as input. The commands indicate that sequences are extracted using bacteria-specific, SSU-specific HMMs (-h HMMs/SSU/bacteria/), regions V1 and V2 are requested (-r V1 -r V2), the long HMM region should be included (-i long), and output files testfileSSUbact.output.csv (-c) and testfileSSUbact.output.fasta (-o) will be generated.

The FASTA file contains all detected regions defined by the "-r" option. In case multiple regions are extracted at once, the FASTA file can be parsed into the individual regions (e.g. using the Unix program grep as "grep -A1 V1 complete.fasta | grep -v -- -- > justV1.fasta"). The script will not detect HMM target regions if these are shorter than the short HMM. A FASTA output for sequence AB033325 using options listed in Figure 2 is shown below. The sequence identifier shows the name of the sequence, the name of the extracted region, the position of the region (start and stop) on the query sequence, and HMM specifications.

>AB033325.1.1478_V1_1_102_includelonghmm
AACGAACGCTGGCGGCATGCCTAACACATGCAAGTCGAACGAGACCTTCGGGTCTAGTGGCGCACGGGTGCGT
AACGCGTGGGAACCTGCCCTTAGGTTCGG
>AB033325.1.1478_V2_85_233_includelonghmm
ACCTGCCCTTAGGTTCGGAATAACTCAGAGAAATTTGAGCTAATACCGGATAATGTCTTCGGACCAAAGATTTA
TCGCCTTTGGATGGGCCCGCGTTGGATTAGCTAGTTGGTGGGGTAAAGGCCTACCAAGGCGACGATCCATAGC
TG

The csv file (Figure 3) indicates the positions of all detected HMM targets as well as the HMM targets that were not detected ("notfound"). In the last columns, absolute positions of all detected V-regions are reported. Special cases such as detection of only one HMM or detection of false-positives are marked in the respective cell. The csv file can be easily imported in programs such as Excel or OpenOffice-Calc.

	A	B	C	D	E	F	G	H	I	J	K
1	Command:	vxtractor.pl -h HMMs/SSU/bacteria/ -r V1 -r V2 -i long -o testfileSSUbact.output.fasta -c testfileSSUbact.output.csv testfileSSUbact.fasta									
2	Options:	Include long HMM in fasta									
3	Sequence	V1leftlong	V1leftshort	V1rightlong	V1rightshort	V2leftlong	V2leftshort	V2rightlong	V2rightshort	V1	V2
4	'AF418950.1	'1-40	'3-40	'notfound	'notfound	'86-136	'86-136	'183-232	'183-232	'notfound	'86-232
5	'AJ292687.1.	'2-40	'3-40	'59-106	'59-80	'89-139	'89-139	'204-253	'204-253	'2-106	'89-253
6	'X81956.1.14	'1-40	'3-40	'64-110	'notfound	'93-143	'93-143	'192-241	'192-241	'1-110 HMM	'93-241
7	'AY218772.1	'1-40	'3-40	'75-122	'75-102	'105-155	'105-155	'226-275	'226-275	'1-122	'105-275
8	'AB088944.1	'1-40	'4-40	'69-114	'69-96	'99-149	'99-149	'214-263	'214-263	'1-114	'99-263
9	'AY278624.1	'1-40	'3-40	'75-122	'75-101	'105-155	'105-155	'220-269	'220-269	'1-122	'105-269
10	'AB200224.1	'1-40	'3-40	'75-122	'75-101	'105-155	'105-155	'220-269	'220-269	'1-122	'105-269
11	'ABJL020000	'1-40	'3-40	'71-118	'71-97	'101-151	'101-151	'215-264	'215-264	'1-118	'101-264
12	'AY218622.1	'1-40	'3-40	'69-115	'69-96	'99-149	'99-149	'213-262	'213-262	'1-115	'99-262
13	'EU421894.1	'1-40	'3-40	'63-110	'63-90	'93-143	'93-143	'208-257	'208-257	'1-110	'93-257
14	'AF050539.1	'1-40	'3-40	'71-118	'71-98	'101-151	'101-151	'216-265	'216-265	'1-118	'101-265
15	'EF509046.1.	'1-38	'3-37	'68-114	'68-93	'97-147	'97-147	'213-262	'213-262	'1-114	'97-262
16	'AY531639.1	'1-40	'3-40	'57-104	'57-84	'87-137	'87-137	'201-250	'201-250	'1-104	'87-250
17	'Y11329.1.14	'1-40	'3-40	'55-100	'55-81	'85-135	'85-135	'196-245	'196-245	'1-100	'85-245
18	'AJ431215.1.	'1-40	'3-40	'75-122	'75-101	'105-155	'105-155	'218-267	'218-267	'1-122	'105-267

Figure 3. Screenshots of a csv output table imported into Excel. Each query sequence is listed in the first column with the positions of the detected regions for each HMM in the following columns and the absolute position of each detected V-region in the last columns.

The script will report special cases. In general, the script can detect five cases, (i) long and short HMM target detected (normal), (ii) only long HMM target detected, (iii) only short HMM target detected, (iv) no HMM target detected, and (v) long and short HMM targets are detected but at different (contradictory) locations. The script will not report messages for cases where long and short HMM targets are found (i). If neither of the HMM targets is detected, the occurrence is marked with "notfound" in the csv table and the particular region cannot be extracted to the FASTA file (iv). If only the long (ii) or the short (iii) HMM target is detected, the script reports these cases in the FASTA header, e.g. with "HMM=longleft" when only the long HMM target was detected. Most importantly, the script will report cases where long and short HMM targets do not match at the same location (v). In this case, the script will trust the long HMM, since these are usually more reliable, and it will mark the sequence header with, e.g., "HMM=mismatchleft". This particular case means that the short HMM target was detected outside (left) of the long HMM target and the region matching the long HMM target is extracted.

The script will detect false-positives as long as their detection is amenable to algorithmic interpretation. There are two main categories. First, for individual regions, false-positives are reported if HMMright is detected to the left (upstream) of HMMleft. The script will mark these occurrences in the csv table and on the screen; the sequence cannot be extracted. As a second measure for false-positives, the user can select the option "-a" in the script in order to check for the correct order of the V-regions. If for example V2 is detected left of V1 then the script will also report these errors in the csv file; however, the regions will be extracted since they are still intact.

5. Discussion

The HMMs were built in order to extract every prominent hypervariable region along the bacterial, archaeal, and fungal SSU and LSU rRNA genes. The locations of the hypervariable regions and the corresponding HMMs can be seen in the sequence variability plots of Appendix A-F. Although the HMMs were tailored to be as unforgiving as possible, it is not a good idea to use whole genomes or very large sequences as input to the script. This increases the chances of false-positives (particularly for the shorter HMMs). Although the script takes measures against false-positives (see above), some cases will probably be impossible to detect in an automated way. The HMMs will also not perform very well on sequences of poor read quality and many IUPAC ambiguities. Poorly read base-pairs are unfortunately common near the 5' and 3' ends of sequences, which can pose a problem if those regions coincide with the location of the present HMMs.

Before you run the script on your data, you should convince yourself that the default settings are suitable for your purposes. Do this by running the script on the first 10 of your sequences and see what comes out. Please take the time to look at how the default settings perform on your sequences! You can compare the extracted regions to the reference alignments provided to decide if the correct regions are being extracted.

The function *hmmscan* uses heuristic filters to increase the search speed. This reduces the power to detect divergent regions. V-Xtractor uses *hmmscan --max* in order to turn off all heuristic filters. This increases detection power but might reduce speed. If speed is preferable over power, the user might want to adjust the script and remove the *--max* option. However, collecting benchmarks on large datasets we found that speed is not an issue.

If you find that the regions are not located when in fact they should have been, you may need to adjust the *hmmscan* e-values and scores to be more allowing. The tailored e-values and scores were hardcoded in the respective HMM files in the line starting with DESC. However, we do not recommend modifying these values without good reasons, as countless hours went into selecting them. In order to find alternate thresholds for your sequences, run *hmmscan --max* by hand for a few of your sequences to see at what e-values and scores the HMMs in question are found, and then change the files accordingly. Note that you may still want the e-values and scores to be as unforgiving as possible in the interest of a low number of false-positives.

Some taxa may have very deviant ribosomal genes (cf. Boucher and Doolittle, Nature 2002), and the script can be expected to perform sub-optimally on these. If you find this to be the case, you may have to compile special HMMs for these only, and use these HMMs only for those taxa. In our experience it is not a good idea to force deviant sequences into the provided HMMs since it will detract from the generality of the HMMs.

Since some regions of the SSU and LSU rRNA genes are highly conserved across the taxonomic domains, certain domain-specific HMMs will extract the target regions from other domains, i.e., a bacterial HMM can sometimes detect the corresponding region in a eukaryote sequence. We tried to minimize this effect by carefully tailoring the HMMs, but the cross-detection among taxonomic lineages is to some extent unavoidable for some highly conserved regions. Therefore, we recommend either using the script on datasets that are domain-specific or processing a mixed dataset with all corresponding HMM sets in turn and parse the output according to taxonomic information. Users only interested in sorting SSU sequences according to domain may find Metaxa (DOI: 10.1007/s10482-011-9598-6) to be of use.

We ask the user to understand that we have made every effort to ensure optimal performance of the HMMs datasets. However, as new entries fill the public databases even better reference alignments can be used to design general HMMs. In our evaluation of 314,032 bacterial 6,185 archaeal and 2,706 fungal SSU sequences, we obtained average extraction efficiencies of 99.9%, 99.7%, and 98.6% respectively.

As for the LSU rRNA gene sequences, the extracted ("hypervariable") regions are here also called V-regions and are numbered V01-V15. They are formally known as D regions in eukaryotes. For the bacterial LSU rRNA genes, extraction of two V-regions – 10 and 11 – is based on one less conserved segment of the bacterial LSU, for lack of very conserved segments in that LSU part (see red arrow in Appendix graph D). They are provided on the understanding that they will not work for all bacterial lineages. If this very region is of major interest to the particular study, it is recommended to extract from 10 through 11 in order not to use this particular segment. We evaluated 20 LSU sequences from each domain. All V-regions from all 20 archaeal LSU sequences were extracted. All V-regions from all 20 bacterial LSU sequences were extracted, except seven V10 and V11 regions. The extractions failed because of the lack of detection of the less conserved HMM located between region V10 and V11 (Appendix D). All V-regions from all 20 fungal LSU sequences were extracted, except region V11 in one of the sequences. The 5' HMM of V11 is very short due to the lack of extended conserved sequence data; thus, extraction of this region might sometimes fail.

For fungal LSU, V01 is the start of the LSU, V02 corresponds to D1, V03 to D2, and so on to V14, which corresponds to D12 (since there are two D7: D7a and D7b; see Nucleic Acids Res. 12 (8), 3563-3583 (1984)). V15 corresponds to the end of the LSU. Thus, V01 and V15 are not formal D regions; they are provided for anyone who also wishes to address the very start, and the very end, of the LSU.

6. Appendix

The graphs below show the sequence variability along the bacterial, archaeal, and fungal SSU and LSU rRNA genes. The red circles represent the frequency of the most common residue at every position in the alignment. The red lines represent the average frequency in 50-nucleotide windows. The green areas represent the location of the HMMs.
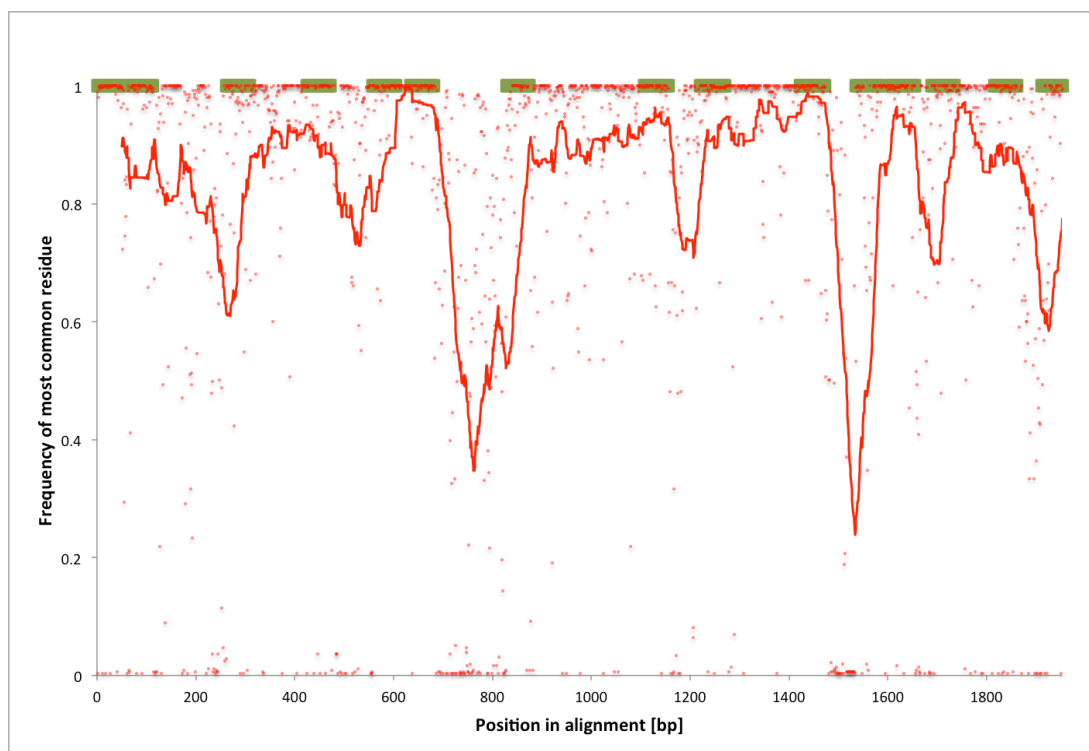
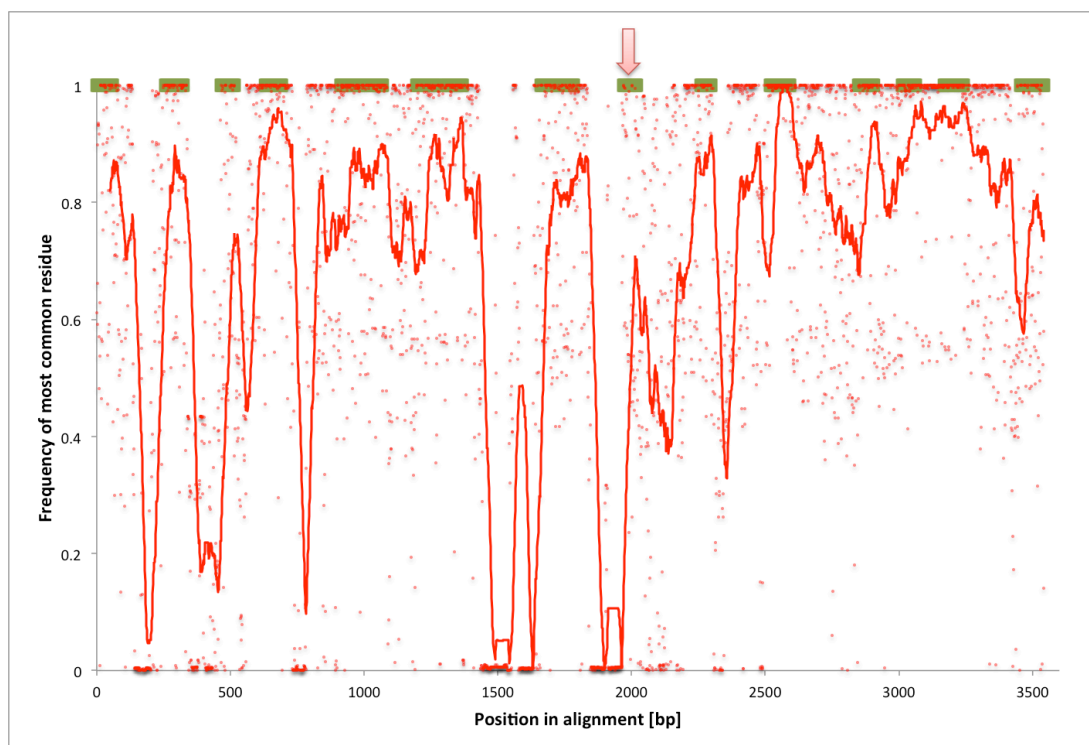*A) Sequence variability plot of 13,104 bacterial SSU rRNA genes*
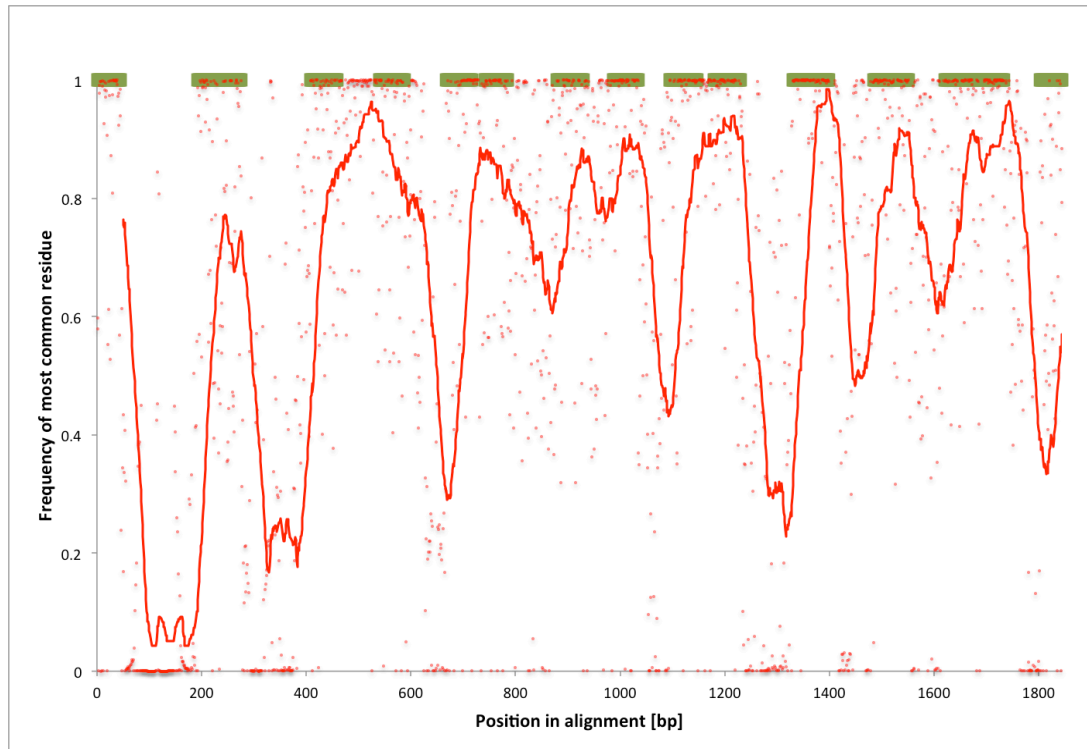
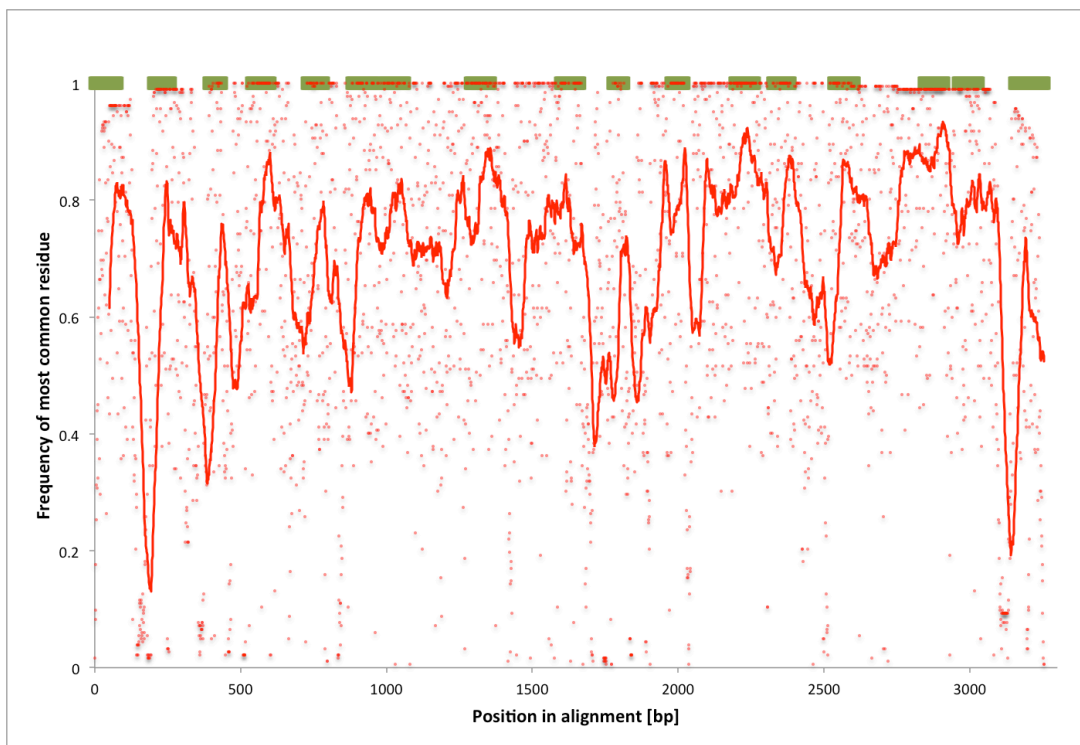*B) Sequence variability plot of 247 archaeal SSU rRNA genes*

9
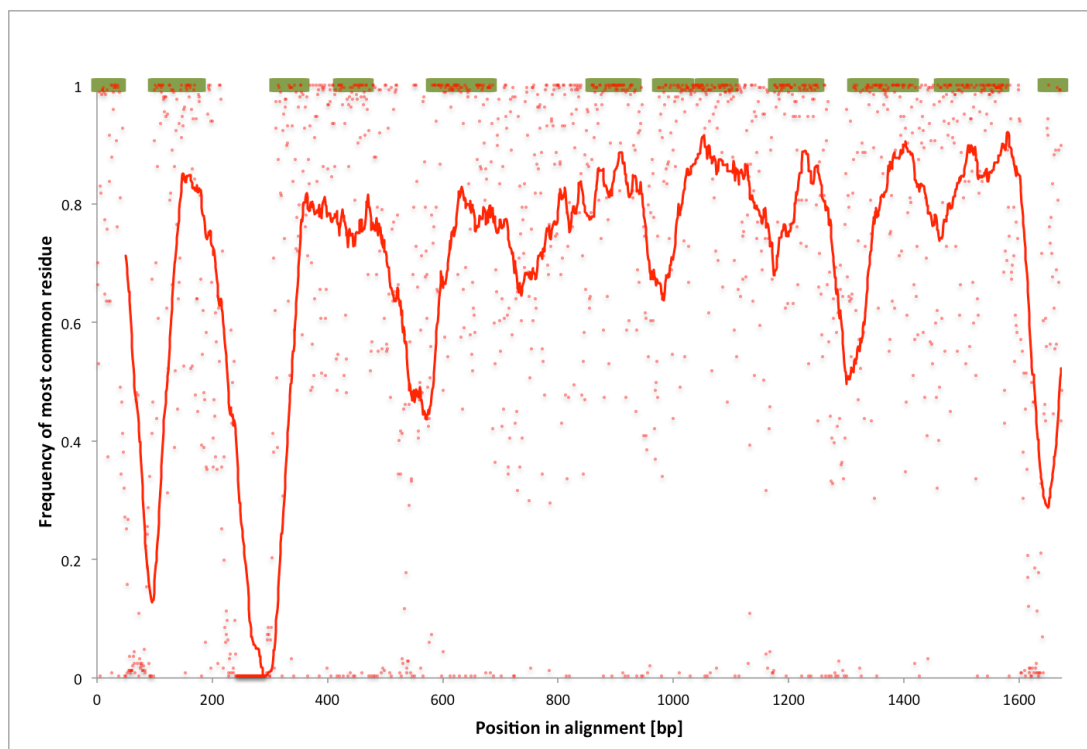
C) Sequence variability plot of 357 fungal SSU rRNA genes

D) Sequence variability plot of 3240 bacterial LSU rRNA genes

E) Sequence variability plot of 182 archaeal LSU rRNA genes

F) Sequence variability plot of 1082 fungal LSU rRNA genes